

# First Step in the Convolutional Neural Networks

## Detailed description of first steps in convolutional neural networks

Yurii Hychka  
Faculty of Electronics  
Ivan Franko National University of Lviv  
Lviv, Ukraine  
e-mail: yuriy.hychka@gmail.com,  
Orcid:0000-0002-3238-8439

**Abstract**—The purpose of this article is to gather and structurize the knowledge about convolutional neural networks and to simplify first steps in this type of networks training.

**Index Terms**—Convolutional Neural Network, ReLU, Pooling, Filtering, Image Recognition;

### I. INTRODUCTION

Today a large number of products and companies are based on CNN: Facebook uses neural networks for automatic comparison between tags and searching users' faces on images, Google uses for image analysis and searching images by description, Amazon – to generate goods recommendation.

CNN have become a breakthrough in the field of computer vision. This is connected with the fact, that CNN is a prototype of human's visual cortex of the brain. In one of David H. Hubel's and Torsten Wiesel's researches it was shown, that some neurons of visual cortex was reacting to the perception of the limits of a particular orientation. David H. Hubel and Torsten Wiesel found out that all these neurons are concentrated in the shape of core architecture and form visual perception. This idea was put into a main principles of CNN functioning.

The advantages of CNN over other classification algorithms are higher speed of input data processing and response. The main difference is the necessity of preliminary network training. However, in serial production of devices or applications that use CNN, one training is enough to determine the necessary parameters of weights and install them in all networks that will solve given task.

### II. THE MAIN STRUCTURE ELEMENTS OF CNN

The simplest NN consists of consecutive stages of convolution, association and classification. These stages are called layers of the neural network.

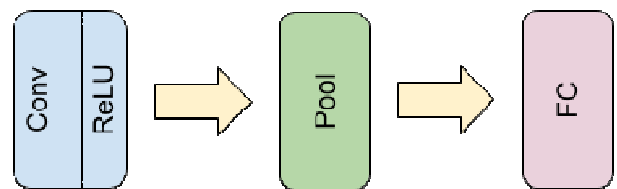


Figure 1. Schematic visualization of Convolutional neural network.

#### A. Convolutional layer

By analogy with the work of the visual cortex of the brain, the convolutional layer of the neural network distinguishes special features of input signal. Generally, this process is to obtain the convolution of input signal and the kernel of the filter. It is possible to use other types of filtering, that will help to obtain additional features, typical for the signal.

#### B. Pooling layer

The purpose of the pulling layer is to reduce matrix dimension, received as a result of overlaying the filters. This step allows to save the features and to reduce calculation performances, needed for network training.

#### C. ReLU layer

The ReLU layer is not always considered as a individual layer. That is because it is used along with convolutional layer. Its purpose is to standardize output of CL. Moreover, the using of ReLU positively affects the reduction of network overfitting effect.

#### D. Fully connected layer

Feature map is received as a result of the functioning of previous 3 layers. For each class of input signals, this map will be significantly different. Fully connected layer is needed to classify received feature map and as a result to show possibilities of input signal to belong to each of the specified class. In general, fully connected layer is used as a classifier.

### III. CONVOLUTIONAL LAYER

CL consists of a great number of filters, which purpose is to distinguish the features of input signal. The results of filtration depend on the size of the filter kernel and how it is initialized.

The complexity of the features that it can distinguish depends on the size of the filter. Also the size of the feature it distinguishes directly depends on the size of the filter. For the MNIST dataset handwriting recognition task, filters with a kernel size of 3x3 or 5x5 will produce a good result, because images in this dataset are 28x28 pixels. The example of handwritten number from the MNIST dataset is shown in figure 2.

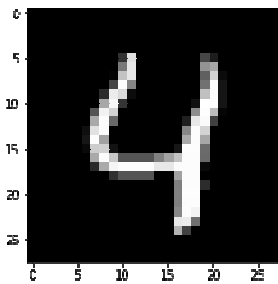
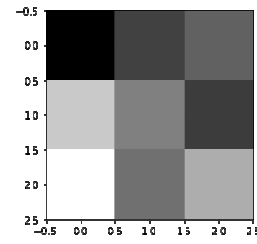


Figure 2. Example of handwritten number from dataset MNIST

#### A. Filter kernel initialization

The number of received signal features and their quality depends on what way of initialization was chosen. As a result, the quality of recognition also depends on that way of initialization. One of the most common methods of filter kernel completion with the values is the application of uniform distribution



$$\begin{bmatrix} -0.74792015 & -0.30615073 & -0.0844353 \\ 0.61618833 & 0.11913138 & -0.33133936 \\ 0.98399669 & 0.01561422 & 0.42898352 \end{bmatrix}$$

Figure 3. Graphical and numerical representation of filter matrix kernel

In the figure 3 is shown randomly generated matrix of a filter core with values within  $[-1; 1]$ .

#### B. Convolution

The process of convolution allows to distinguish the signal features.

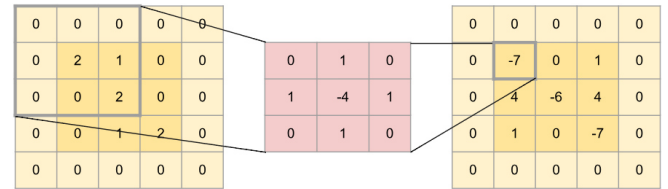


Figure 4. Visualization of convolutional process

For each element of input signal is given a window with the size  $n \times n$ , where  $n$  is the size of the filter kernel. This matrix is filled with the elements adjacent to the given element in such a way, that selected element is on the center. Afterwards, dot production of received matrix by matrix kernel has to be done. The example of this process is shown in figure 4. Convolution is calculated by formula (1)

$$V_{lk} = \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{ij} \quad (1)$$

where:

$V_{lk}$  - output value of selected element

$f_{ij}$  - value of filter kernel element

$d_{ij}$  - value of window element

As a result, special features are distinguished from the input signal, as it is shown in the figure 5.

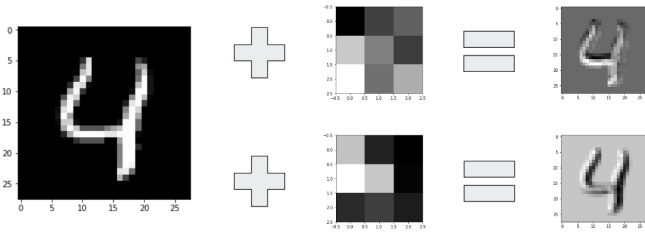


Figure 5. Result of two random kernel filter convolution with image

#### IV. POOLING LAYER

Pooling layer allows to reduce matrix dimension by selecting elements from full matrix to the new matrix due to certain criterion.

The principle of that process is a window selection from the full matrix and applying aggregation functions. The window is shifted with a certain step, and the results of the aggregation function will be the values of the reduced matrix.

Typically, the  $\max()$  function is used as an aggregation function. The using of this function allows to distinguish the most important features and not to use unimportant. The figure 6 shows an explicit explanation of the results of this function. In the example, a window is used in the dimension of  $2 \times 2$  with a step 2. The input matrix has dimensions  $4 \times 4$ , and after pooling its dimension falls to  $2 \times 2$ .

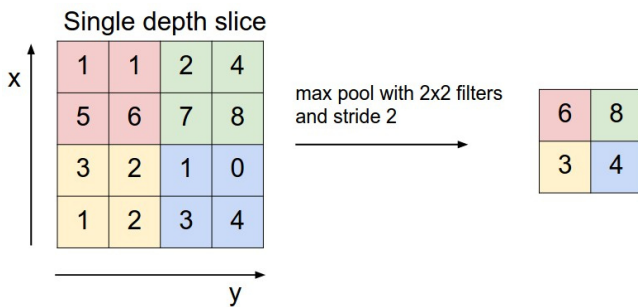


Figure 6. Example of convolution with using  $\max()$  aggregation function

Also, the pooling layer is a good tool for protection against overfitting.

Figure 7 shows the result of pooling, that was applied on 2 prefiltered images.

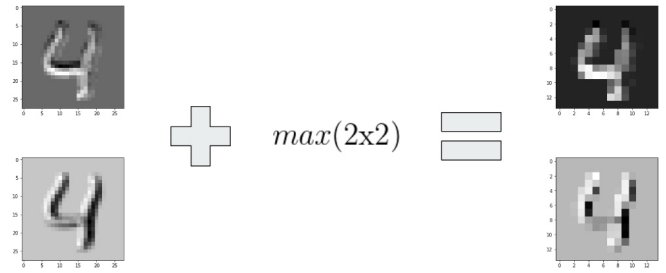


Figure 7. Result of pooling applying

#### V. RELU LAYER

Relu layer is used for modelling output neuron. Besides Relu function, modelling can be done by using non-linear non-saturating functions like  $\tanh()$  or logistic curve. But comparing to non-linear non-saturating ReLU function, using these functions needs more time and epochs of training because of their computable complication and saturation zone.[2]

Функція ReLU описується формулою (2)

$$\text{ReLU} = \max(0, x) \quad (2)$$

where  $x$  - input value of function

The figure 8 shows result of applying ReLU function.

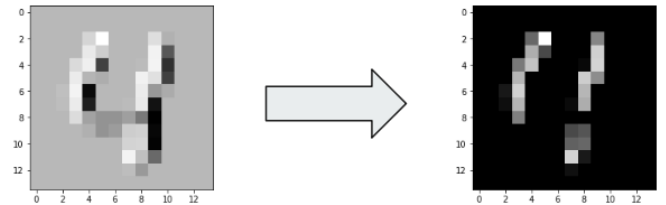


Figure 8. Result of using ReLU function

#### VI. CONVOLUTIONAL NEURAL NETWORK ARCHITECTURES

The simplest CNN consists of one convolutional layer, one pooling layer and one fully connected layer. Application of 6 to 10 filters for CL is enough to obtain effective training while simple tasks. Fig 9. shows CNN scheme. Network matrices dimension in that scheme is made for input date matrix with the dimension  $28 \times 28$ , this is characteristic of MNIST dataset.

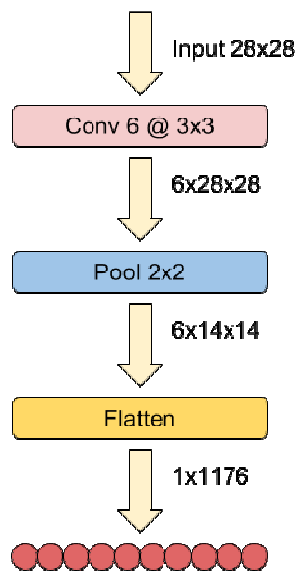


Figure 9. Schematic visualization of Convolutional Neural Network

The "Flatten" step in this scheme is a function that reduces the n-dimensional matrix to the vector. As a result of using this function, a vector of features is obtained. In this case, CNN consists of 10 neurons. Each of them is responsible for one out of 10 classes.

Beside this simple architecture there are a lot more, that deserve attention. The main are AlexNet, VGG та ResNet.

## CONCLUSION

This article contains simplified description of the main components of CNN and the simplest CNN scheme. This work can be used for pedagogical purposes to explain and demonstrate the principles of CNN functioning. Moreover, this work can be used to structure and consolidate gained knowledge.

## REFERENCES

- [1] Keras Documentation. Retrieved July 9, 2018, from <https://keras.io/>
- [2] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [3] CS231n Convolutional Neural Networks for Visual Recognition. Retrieved July 9, 2018, from <http://cs231n.github.io/convolutional-networks/#norm>
- [4] LeCun, Yann; Corinna Cortes; Christopher J.C. Burges. "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges". Retrieved July 9, 2018, from <http://yann.lecun.com/exdb/mnist/>